

## Project 3 Process Scheduling

For this third project I put all code into a single .java file, where I coded the round robin method into the main class, and coded two separate classes for FCFS(first come first serve) and SJF(shortest job first). But those methods are called from main so when the file is run all three methods are called one after the other and they print out their results in three "tables" of results.

Execution: the one issue that might come up when running this file is that I am not certain if the pXX.txt files are going to be searched for in the proper place if the .java file is pulled out of the Homework3 folder. The txt files are located in the folder alongside the .classpath, and .project files as it seems that is the default location eclipse cares to look at. I couldn't find another way of coding this section so I hope it causes no issues.

Results: I excluded the critical path portion of the coding since you mentioned the critical path numbers would be 0 for this project. So in a way it would mean that critical path doesn't entirely matter, plus it would only affect round robin anyway. But when I ran my program with my 5 test txt files I got the following output:

5 processes where found

STARTING SJF:

Process #4 finished in 500ms

Process #2 finished in 1165ms

Process #3 finished in 1885ms

Process #1 finished in 4785ms

Process #0 finished in 12000ms

Done with SJF

STARTING FIRST COME FIRST SERVE:

Process #0 finished in 7215ms

Process #1 finished in 10115ms

Process #2 finished in 10780ms

Process #3 finished in 11500ms

Process #4 finished in 12000ms

Done with FCFS

STARTING ROUND ROBIN

Process #4 is finished, elapsed time:2500

Process #2 is finished, elapsed time:3665

Process #3 is finished, elapsed time:3885

Process #1 is finished, elapsed time:7785

Process #0 is finished, elapsed time:12000

As we all knew to expect they are all finishing up at the same time since critical path isn't there to clog up round robin. But SJF was the scheduling process that finished up with a process faster, as expected, showing it favors small processes first and just concentrates on getting them out of the way. Also it may seem that FCFS took the longest to get a job out but that was entirely because the sample txt files started off with the longest process I had as p00.txt so the results represents what could happen if a huge file were to come into the queue first and clog it up for all the tiny processes to come.